Algorithm for Fuzzy Maximum Flow Problem in Hyper-Network Setting

Linli Zhu[1,*], Xiaozhong Min[1], Wei Gao[2] and Haixu Xi[1]

[1]*School of Computer Engineering, Jiangsu University of Technology, Changzhou, Jiangsu 213001, P.R. China;* [2]*School of Information, Yunnan Normal University, Kunming, Yunnan 650500, P.R. China*

Abstract: Maximum flow problem on hypergraphs (hyper-networks) is an extension of maximum flow problem on normal graphs. In this paper, we consider a generalized fuzzy version of maximum flow problem in hyper-networks setting. Our algorithm is a class of genetic algorithms and based on genetic tricks. The crisp equivalents of fuzzy chance constraints in hyper-networks setting are defined, and the execution steps of encoding and decoding are presented. Finally, we manifest the implement procedure.

Keywords: Coding technology, genetic algorithm, hypergraph, maximum flow problem.

1. INTRODUCTION

Maximum flow problem of weighted graph, an important component of graph theory and artificial intelligence, has been widely used in many fields, such as computer network, data mining, image segmentation and ontology computation (see [1-7]). Hyper-graph is a subset system for limited set, which is the most general discrete structure, and it is the generalization of the common graph. For many practical problems, adopting the concept of hyper-graph is more usefully than adopting the concept of graph. At present, the model of hypergraph has been applied in many fields, such as: VLSI layout, electricity network topology analysis. Recently, intelligence algorithms and learning algorithms on hyper-graph and its computer applications are studied by researchers (see [8-17] for example).

Let $V=\{v_1,v_2,\ldots,v_m\}$ be a limited set, E is family of subset of V, *i.e.*, $E\subseteq 2^V$. Then $H=(V,E)$ is a hypergraph on V. the element of V is called a vertex, the elements of E is called a hyperedge. Let $|V|$ be the order of H, $|E|$ be the scale of H. $|e|$ is basic number of hyperedge e. $r(H)=\max_j |e_j|$ is rank of hyperedge e, and $s(H)=\min_j |e_j|$ lower rank of hyperedge e. If $|e|=k$ for each hyperedge e of E (that is $r(H)=s(H)=k$), then H is a k-uniform hypergraph. If $k=2$, then H is just a normal graph.

A hypergraph H is called a simple hypergraph or a sperner hypergraph, if any two hyperedges are not contained with each other. Let $H'=(V,E')$ is a hypergraph on V, if $E'\subset E$, then H' is a part-hypergraph of H. For $S\subseteq V$, $H[S]=\{e\in E:e\subseteq S\}$ is called a sub-hypergraph of H induced by S.

Hypergraph H can be represented by graph by using the set of vertices to represent the elements of V. If $|e_j|=2$, using a continuous curve which attach to the elements of e_j to representing e_j; If $|e_j|=1$, using a loop which contain e_j to represent e_j; If $|e_j|\geq 3$, using a simple close curve which contains all the elements of e_j to represent e_j.

In this paper, we assume H is a weighted hypergraph, each edge given a wight $w(e)$. The degree of vertex v_j in hypergraph H is denoted as

$$\deg_j(H)=\sum_{e\in E} w(e)h(v,e),$$

where

$$h(v,e)=\begin{cases}1, & if \quad v\in e\\ 0, & if \quad v\notin e\end{cases}.$$

Let $\delta(e)=\sum_{v\in V} h(v,e)$. Then, the normalized laplacian $L(H)\in \mathbb{R}^{m\times m}$ on hypergraph H is defined by

$$L_{ij}(H)=\begin{cases}-\sum_{\{i,j\}\subseteq e} w(e)\dfrac{1}{\delta(e)} & i\neq j\\ \deg_j(H) & \text{otherwise}\end{cases}.$$

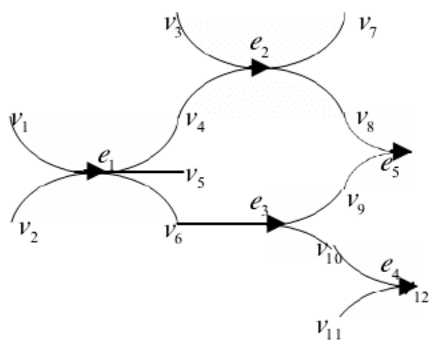
Let $H=(V, E)$ be a fixed a directed, weighted hyper-graph with n vertices which express a hyper-network. In many projects like large super-network research, database systems research, timing research, circuit design research and so on, directed hypergraph models can represent relationships between elements there. Due to its good application background, directed hypergraph theory has become a rapidly developing subject in the field of graph theory.

Specifically, a directed hyper-graph is a hyper-graph where each hyper-edge divided into two sets: $e=(X,Y)$ with $X \cap Y = \varnothing$ and X, Y can be the empty set. Here, X called a tail point set and Y called a head point set denoted by $T(e)$ and $H(e)$ respectively. Similar as undirected hyper-graph, we can define the hyper-road, hyper-path, hyper-cycle in the directed hyper-graph in directed hypernetworks.

We intordunce a {-1,0,1} incidence matrix to represent the directed hyper-graph. The j-th column express the j-th vertex v_j and i-th row express the i-th hyper-edge e_i :

$$[a_{ij}]_{m\times n} = \begin{cases} -1, & v_i \in T(e_j) \\ 1, & v_i \in H(e_j) \\ 0, & \text{otherwise} \end{cases} .$$

Following is an example of directed hyper-graph and its incidence matrix:



	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}
e_1	−1	−1	0	1	1	1	0	0	0	0	0	0
e_2	0	0	−1	−1	0	0	1	1	0	0	0	0
e_3	0	0	0	0	0	−1	0	0	−1	1	0	0
e_4	0	0	0	0	0	0	0	0	0	−1	−1	1
e_5	0	0	0	0	0	0	0	−1	−1	0	0	0

In many hyper-networks applications, there are exist the uncertain factors which can't expressed by fixed functions or parameters. Hence, the fuzzy theory is widely applied in networks and hyper-networks (see [18-22]). In this paper, we consider the fuzzy maximum flow problem in hyper-networks. The new optimization model is presented by virtue of fuzzy capacities calculating and crisp equivalents of fuzzy chance constraints.

2. SETTING

Consider a directed flow hyper- network $H=(V,E,C)$, where V implies the finite set of vertices, denoted by the number $\{1, 2,\ldots, n\}$. E expresses the set of directed hyper-edge, each directed hyper-edge e is denoted by an ordered pair $(H(e),T(e))$, where $e\in E$. C represents the set of directed hyper-edge capacities. In the fuzzy maximum flow problem in hyper-networks setting, every directed hyper-edge e has a nonnegative, independent, fuzzy flow capacity ξ_e with the membership functions μ_e. Then, for each pair of vertices (v_i, v_j), we use

$$\xi_{ij} = \sum_{\{v_i,v_j\}\subset e\in E} \xi_e$$

to denote its fuzzy flow capacity associated with certain membership functions μ .

In what follows, flow representation is employed by:

$$\boldsymbol{x} = \{x_{ij} = \sum_{\{v_i,v_j\}\subset e\in E} x_e\}$$

where x_e denotes the flow of directed hyper-edge e. The flow is called a feasible flow in hyper-networks setting if the below two conditions are established:

(1) For each vertex, the outgoing flow and incoming flow must meet the following balance conditions.

$$\begin{cases} \sum_{\{v_1,v_j\}\subset e\in E} \boldsymbol{x}_{1j} - \sum_{\{v_j,v_1\}\subset e\in E} \boldsymbol{x}_{j1} = f \\ \sum_{\{v_i,v_j\}\subset e\in E} \boldsymbol{x}_{ij} - \sum_{\{v_j,v_i\}\subset e\in E} \boldsymbol{x}_{ji} = 0, \quad 2 \le i \le n-1 \\ \sum_{\{v_n,v_j\}\subset e\in E} \boldsymbol{x}_{nj} - \sum_{\{v_j,v_n\}\subset e\in E} \boldsymbol{x}_{jn} = -f \\ \forall e\in E \end{cases}$$

in which f denotes the flow of the hyper-network H.

(2) The flow at each directed hyper-edge must be satisfied by the capacity constraint.

In this paper, we use the fuzzy set technologies to deal with the fuzziness, which were first introduced by Zadeh. In fuzzy setting, there are three classes of measures consisting of necessary, possibility and credibility measure [23, 24]. As we know, a fuzzy event may fail even though its possibility attains 1, and established even though its possibility reaches 0. However, the fuzzy event should be happened when its credibility becomes 1 and fail when its credibility is zero. In our article, we model fuzzy maximum flow problem in hyper-network setting in terms of credibility measure. Our technologies mainly followed the tricks raised in [25].

Use ξ to denote the fuzzy variable with the membership function $\mu(x)$. Hence, the credibility measure (Cr), the necessity measure (Nec), and the possibility measure (Pos) of the fuzzy event $\{\xi \ge r\}$ can be denoted by

$$\text{Cr}\{\xi \ge r\} = \frac{1}{2}[\text{Pos}\{\xi \ge r\} + \text{Nec}\{\xi \ge r\}],$$

$$\mathrm{Nec}\{\xi \ge r\} = 1 - \sup_{u<r} \mu(u)$$

and

$$\mathrm{Pos}\{\xi \ge r\} = \sup_{u\ge r} \mu(u),$$

respectively

In several applications, the experts are interested in the hyper-networks flow which meet certain chance constraints with at least some fixed confidence level α. A flow x is called the α-optimistic maximum flow (α-OMF) from vertices v_1 to v_n if (see [25]):

$$\max\{f|\mathrm{Cr}\{\xi \ge \boldsymbol{x}\} \ge \alpha\}$$

$$\ge \max\{f'|\mathrm{Cr}\{\xi \ge \boldsymbol{x}'\} \ge \alpha\}$$

for any flow $\boldsymbol{x}'$ from vertices v_1 to v_n, and α here is implied as a predetermined confidence level.

Chance-constrained programming provides us a useful tools for modelling fuzzy decision systems [26-30]. The basic idea of chance-constrained programming of fuzzy maximum flow problem in hyper-networks setting is to optimize the flow value of hyper-network with some confidence level subject to certain chance constraints. For searching the α-OMF in hyper-networks setting, we raise the following model.

$$\begin{cases} \max f \\ \text{s. t.:} \\ \sum\limits_{\{v_1,v_j\}\subset e\in E} \boldsymbol{x}_{1j} - \sum\limits_{\{v_j,v_1\}\subset e\in E} \boldsymbol{x}_{j1} = f \\ \sum\limits_{\{v_i,v_j\}\subset e\in E} \boldsymbol{x}_{ij} - \sum\limits_{\{v_j,v_i\}\subset e\in E} \boldsymbol{x}_{ji} = 0, \quad 2 \le i \le n-1 \\ \sum\limits_{\{v_n,v_j\}\subset e\in E} \boldsymbol{x}_{nj} - \sum\limits_{\{v_j,v_n\}\subset e\in E} \boldsymbol{x}_{jn} = -f \\ \mathrm{Cr}\{\xi_{ij} \ge \boldsymbol{x}_{ij}\} \ge \alpha \quad \text{for each pair of } (v_i,v_j) \\ f \ge 0 \end{cases} \qquad (1)$$

where α is a predetermined confidence level supplied as an appropriate margin *via* the field experts.

3. ALGORITHM FOR FUZZY MAXIMUM FLOW PROBLEM IN HYPER-NETWORKS SETTING

A popular technology for solving fuzzy chance-constrained programming model is to convert he chance constraint

$$\mathrm{Cr}\{\xi \ge \boldsymbol{x}\} \ge \alpha$$

into its crisp equivalent and thus solve the equivalent crisp model in deterministic environment. In our hyper-network setting, we suppose that ξ are general fuzzy variables with membership functions $\mu_\xi(x)$. Then, we infer that $\mathrm{Cr}\{\xi \ge \boldsymbol{x}\} \ge \alpha$ if and only if $\boldsymbol{x} \le K_\alpha$ with

$$K_\alpha = \begin{cases} \sup\{K \mid K = \mu^{-1}(2\alpha)\}, & \text{if } \alpha < \frac{1}{2} \\ \inf\{K \mid K = \mu^{-1}(2(1-\alpha))\}, & \text{if } \alpha \ge \frac{1}{2} \end{cases}.$$

Suppose that $\xi_{ij} = \sum_{\{v_i,v_j\}\subset e\in E} \xi_e$ are general fuzzy variables with membership functions $\mu_{\xi_{ij}}(x) = \sum_{\{v_i,v_j\}\subset e\in E} \mu_{\xi_e}(x)$ respectively. Thus, the optimization model (1) can be reformulated as follows:

$$\begin{cases} \max f \\ \text{s. t.:} \\ \sum\limits_{\{v_1,v_j\}\subset e\in E} \boldsymbol{x}_{1j} - \sum\limits_{\{v_j,v_1\}\subset e\in E} \boldsymbol{x}_{j1} = f \\ \sum\limits_{\{v_i,v_j\}\subset e\in E} \boldsymbol{x}_{ij} - \sum\limits_{\{v_j,v_i\}\subset e\in E} \boldsymbol{x}_{ji} = 0, \quad 2 \le i \le n-1 \\ \sum\limits_{\{v_n,v_j\}\subset e\in E} \boldsymbol{x}_{nj} - \sum\limits_{\{v_j,v_n\}\subset e\in E} \boldsymbol{x}_{jn} = -f \\ \boldsymbol{x} \le K_{\alpha_{ij}} \\ f \ge 0 \end{cases} \qquad (2)$$

where

$$K_{\alpha_{ij}} = \begin{cases} \sup\{K \mid K = \mu^{-1}(2\alpha_{ij})\}, & \text{if } \alpha_{ij} < \frac{1}{2} \\ \inf\{K \mid K = \mu^{-1}(2(1-\alpha_{ij}))\}, & \text{if } \alpha_{ij} \ge \frac{1}{2} \end{cases}.$$

The directed hyper-edge capacities of a hyper-network are independent trapezoidal fuzzy variables denoted as $\xi_{ij} = (a_{ij}, b_{ij}, c_{ij}, d_{ij})$, respectively. Therefore, if $\alpha > 0.5$, the model (1) can be further expressed as the following version:

$$\begin{cases} \max f \\ \text{s. t.:} \\ \sum\limits_{\{v_1,v_j\}\subset e\in E} \boldsymbol{x}_{1j} - \sum\limits_{\{v_j,v_1\}\subset e\in E} \boldsymbol{x}_{j1} = f \\ \sum\limits_{\{v_i,v_j\}\subset e\in E} \boldsymbol{x}_{ij} - \sum\limits_{\{v_j,v_i\}\subset e\in E} \boldsymbol{x}_{ji} = 0, \quad 2 \le i \le n-1 \\ \sum\limits_{\{v_n,v_j\}\subset e\in E} \boldsymbol{x}_{nj} - \sum\limits_{\{v_j,v_n\}\subset e\in E} \boldsymbol{x}_{jn} = -f \\ \boldsymbol{x}_{ij} < (2\alpha-1)a_{ij} + 2(1-\alpha)b_{ij} \\ f \ge 0 \end{cases} \qquad (3)$$

Next, we focus on the genetic algorithm which was introduced by Holland [31] to optimal the combinatorial problems. Several results on genetic algorithm can refer to [32-35].

Here, we use priority-based encoding tricks for our fuzzy maximum flow problem in hyper-network setting. We encode a chromosome in terms of obtaining each vertex a distinct priority number from 1 to n. Fig. (**1**) show an example. The hyper-path from 1 to n is determined by continuously adding the useful vertex with the highest priority into the hyper-path until the hyper-path arrives the terminal vertex in hyper-networks. Furthermore, we decode it into a flow in the hyper-network by hyper-path algorithm by the below decoding technology.

Position: vertex ID	1	2	3	4	5	6	7	8	9	10
value priority	7	3	10	4	2	5	9	6	1	8

Fig. (1). Encoding operation.

For searching the flow of hyper-network, we infer the below procedure where l denotes the number of hyper-paths, p_l implies the l-th hyper-path from vertex 1 to n, f_l expresses the flow on this hyper-path, $c_{ij} = \sum_{\{v_i, v_j\} \subset e \in E} c_e$ denotes the capacity sum for each pair of vertices (v_i, v_j), N_i represents the set of vertices with all vertices adjacent to vertex v_i.

Step 1. Mark the number of hyper-paths $l \leftarrow 0$.

Step 2. If $N_1 \neq \varnothing$, then $l \leftarrow l+1$; otherwise, go to step 8.

Step 3. The hyper-path p_l is constructed by adding the useful vertex with the highest priority into the hyper-path until the hyper-path arrives the terminal vertex. Choose the sink vertex a of hyper-path p_l.

Step 4. If the sink vertex a=n, continue; otherwise, update the set of vertex N_i such that $N_i = N_i - \{a\}$, then go back to step 2.

Step 5. Determine the flow f_l of the hyper-path p_l in view of $f_l \leftarrow f_{l-1} + \min\{c_{ij} \mid \{v_i v_j\} \subset e \in p_l\}$.

Step 6. Implement the flow capacity c_{ij} of each directed hyper-edge update and each pair of vertices (v_i, v_j). Take a new flow capacity c_{ij}^- using the formula

$$c_{ij}^- = c_{ij} - \min\{c_{ij} \mid \{v_i v_j\} \subset e \in p_l\}.$$

Step 7. If the flow capacity $c_{ij} = 0$, implement the set of vertex N_i update such that the vertex j adjacent to vertex i, $N_i = N_i - j$, $\{v_i v_j\} \subset e \in p_l$ and $c_{ij} = 0$.

Step 8. Output the hyper-network flow f_l of this chromosome.

Here, we need the position-based crossover operator which was introduced in the genetic algorithms. An example with 10 vertices is presented in Fig. (**2**).

parent 1. 3 1 2 4 5 8 9 10 7 6
↓ ↓ ↓ ↓
child 3 6 2 4 5 7 9 10 1 8
↗↗ ↗ ↑ ↗ ↑
parent 2 6 2 9 5 4 7 3 1 10 8

Fig. (2). Crossover operation.

The mutation operation is determined *via* exchanging the priority values of two randomly generalized vertices which was expressed in Fig. (**3**).

↓ ↓
parent 3 1 2 4 5 8 9 10 7 6
child 3 1 10 4 5 8 9 2 7 6

Fig. (3). Mutation operator.

We now present our main genetic algorithm for fuzzy maximum flow problem in hyper-network setting.

Step 1. Set genetic parameters by field experts.

Step 2. Initialize pop size chromosomes P_k, k=1, 2, …, pop size.

Step 3. Search the flow for all chromosomes by above procedure, respectively.

Step 4. Calculate the fitness for each chromosome. The evaluation function rely heavily on ranking technology which is denoted by

$$\text{Eval}(P_i) = a(1-a)^{i-1}, \ i=1,2,\ldots, \text{pop size}.$$

where the chromosomes are supposed to have been ranked from good to bad based on their objective scores and $a \in (0, 1)$ is a parameter in the genetic system.

Step 5. Choose the chromosomes for a new population.

Step 6. Update the chromosomes P_k, k=1, 2, …, pop size by virtue of mutation operation and crossover operation technologies presented above.

Step 7. Repeat the 4-6 steps for a fixed number of hyper-cycles.

Step 8. Repeat the maximum flow in this hyper-network.

CONCLUSION

In our paper, we consider the fuzzy maximum flow problem in hyper-networks setting. Our algorithm is designed based on genetic technology and coding theory. The result achieved in our paper illustrates the promising application prospects for algorithms using hypergraph model.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

We thank the reviewers for their constructive comments in improving the quality of this paper. We also would like to thank the anonymous referees for providing us with constructive comments and suggestions. We wish to acknowledge the National Natural Science Foundation of China (61142007) and the general program of Jiangsu University of Technology in China (KYY14013).

REFERENCES

[1] A. Hafezalkotob, and A. M. Ahmad, "Cooperative maximum-flow problem under uncertainty in logistic networks", *Applied Mathematics and Computation*, vol. 250, pp. 593-604, 2015.

[2] K. M. Sullivan, and J. C. Smith, "Exact algorithms for solving a euclidean maximum flow network interdiction problem", *Networks*, vol. 64, no. 2, pp. 109-124, 2014.

[3] S. W. Han, Z. X. Peng, and S. Q. Wang, "The maximum flow problem of uncertain network", *Information Sciences*, vol. 265, pp. 167-175, 2014.

[4] H. Amankwah, T. Larsson, and B. Textorius, "A maximum flow formulation of a multi-period open-pit mining problem", *Operational Research*, vol. 14, no. 1, pp. 1-10, 2014.

[5] M. Gamst, "A local search heuristic for the multi-commodity k-splittable maximum flow problem", *Optimization Letters*, vol. 8, no. 3, pp. 919-937, 2014.

[6] D. Dressler, and M. Strehler, "Polynomial-time algorithms for special cases of the maximum confluent flow problem", *Discrete Applied Mathematics*,vol. 163, pp. 142-154, 2014.

[7] T. Illes, and R. Molnar-Szipai, "On strongly polynomial variants of the MBU-simplex algorithm for a maximum flow problem with non-zero lower bounds", *Optimization*, vol. 63, no. 1, pp. 39-47, 2014.

[8] D. Zhou, J. Huang, and B. Scholkopf, *Beyond Pair-Wise Classification and Clustering using Hypergraphs*. University of Waterloo, Canada, 2005.

[9] D. Zhou, J. Huang, and B. Scholkopf, "Learning with hypergraphs, clustering, classification, and embedding", In: *Proceedings of 20th Annual Conference on Neural Information Processing Systems,* 2006, Vancouver / Whistler, IEEE, Canada, 2006, pp. 1601-1608.

[10] S. Liang, S. Ji, and J. Ye, "Hypergraph spectral learning for multi-label classification", In: *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, ACM, USA, 2008, pp. 668- 676.

[11] G. Chen, J. Zhang, and F. Wang, "Efficient multi- label classification with hypergraph regularization", In: *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, IEEE, USA, 2009, pp. 1658-1665.

[12] L. Yan, J. S. Li, and W. Gao, "Notes on vertex classification problem in hypergraph setting", *Information Technology Journal*, vol. 12, no. 15, pp. 3309-3314, 2013.

[13] L. L. Zhu, and W. Gao, "Analysis of classification algorithm on hypergraph", *The Open Cybernetics & Systemics Journal*, vol. 8, pp. 122-127, 2014.

[14] L. L. Zhu, Y. Gao, and W. Gao, "Arriving and convert times in hyper-networks", *Scholars Journal of Physics, Mathematics and Statistics*, vol. 1, no. 2, pp. 74-80, 2014.

[15] W. Gao, and L. Liang, "Ontology concept similarity computation based on regularization framework of hypergraph", *Microelectronics and Computer*, vol. 28, no. 5, pp. 15-17, 2011.

[16] D. Tisza, A. Olah, and J. Levendovszky, "Novel algorithms for quadratic programming by using hypergraph representations", *Wireless Personal Communications*, vol. 77, no. 3, pp. 2305-2339, 2014.

[17] D. S. Kostyanoi, A. V. Mokryakov, and V. I. Tsurkov, "Hypergraph recovery algorithms from a given vector of vertex degrees", *Journal of Computer and Systems Sciences International*, vol. 53, no. 4, pp. 511-516, 2014.

[18] J. Wang, C. H. Wang, and C. L. P. Chen, "The bounded capacity of fuzzy neural networks (FNNs) *via* a new fully connected neural fuzzy inference system (F-Confis) with its applications", *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 6, pp. 1373-1386, 2014.

[19] H. H. Turan, N. Kasap, and H. Savran, "Using heuristic algorithms for capacity leasing and task allocation issues in telecommunication networks under fuzzy quality of service constraints", *International Journal of Systems Science*, vol. 45, no. 3, pp. 307-324, 2014.

[20] M. Aminnayeri, and P. Biukaghazadeh, "Robust clustering approach for the shortest path problem on finite capacity fuzzy queuing network", *International Journal of Advanced Manufacturing Technology*, vol. 61, no. 5-8, pp. 745-755, 2012.

[21] S. Soudmand, M. Ghatee, and S. M. Hashemi, "SA-IP method for congestion pricing based on level of service in urban network under fuzzy conditions", *International Journal of Civil Engieering*, vol. 11, no. 4A, pp. 281-291, 2013.

[22] T. C. Havens, J. C. Bezdek, and C. Leckie, "A soft modularity function for detecting fuzzy communities in social networks", *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 16, pp. 1170-1175, 2013.

[23] B. Liu, *Theory and Practice of Uncertain Programming*, Physica-Verlag, Heidelberg, 2002.

[24] B. Liu, *Uncertainty Theory: An Introduction To Its Axiomatic Foundations*, Springer-Verlag, Berlin, 2004.

[25] X. Y. Ji, L. X. Yang, and Z. Shao, "Chance constrained maximum flow problem with fuzzy arc capacities", In: *ICIC, 2006, LNAI 4114*, Kunming, China, 2006, pp. 11-19.

[26] N. Zhang, Z. G. Hu, and X. Han, "A fuzzy chance-constrained program for unit commitment problem considering demand response, electric vehicle and wind power", *International Journal of Electrical Power & Energy Systems*, vol. 65, pp. 201-209, 2015.

[27] C. Dai, Y. P. Cai, and Y. P. Li, "Optimal strategies for carbon capture, utilization and storage based on an inexact m(lambda)-measure fuzzy chance-constrained programming", *Energy*, vol. 78, pp. 465-478, 2014.

[28] Y. P. Li, N. Zhang, G. H. Huang, and J. Liu, "Coupling fuzzy-chance constrained program with minimax regret analysis for water quality management", *Stochastic Environmental Research and Risk Assessment*, vol. 28, no. 7, pp. 1769-1784, 2014.

[29] P. Guo, X. H. Chen, M. Li, and J. B. Li, "Fuzzy chance-constrained linear fractional programming approach for optimal water allocation", *Stochastic Environmental Research and Risk Assessment*, vol. 28, no. 6, pp. 1601-1612, 2014.

[30] P. Dursun, and E. Bozdag, "Chance constrained programming models for constrained shortest path problem with fuzzy parameters", *Journal of Multiple-valued Logic and Soft Computing*, vol. 22, no. 4-6, pp. 599-618, 2014.

[31] J. Holland, *Adaptatin In Natural and Artificial System*, University of Michigan Press, Ann Arbor, MI, 1975.

[32] E. D. Howe, and O. Nicolis, "Genetic algorithm in the wavelet domain for large p small n regression", *Communications in Statistics-Simulation and Computation*, vol. 44, no. 5, pp. 1144-1157, 2015.

[33] R. Rashedi, and T. Hegazy, "Capital renewal optimisation for large-scale infrastructure networks: genetic algorithms versus advanced mathematical tools", *Structure and Infrastructure Engineering*, vol. 11, no. 3, pp. 253-262, 2015.

[34] C. Changdar, G. S. Mahapatra, and R. K. Pal, "An improved genetic algorithm based approach to solve constrained knapsack problem in fuzzy environment", *Expert Systems with Applications*, vol. 42, no. 4, pp. 2276-2286, 2015.

[35] S. Karakatic, and V. Podgorelec, "A survey of genetic algorithms for solving multi depot vehicle routing problem", *Applied Soft Computing*, vol. 27, pp. 519-532, 2015.

Received: June 10, 2015 Revised: July 29, 2015 Accepted: August 15, 2015